

або викликати об'єкти по іменах їхніх класів. Сучасні обфускатори дозволяють зберегти зазначені класи від перейменування, однак подібні обмеження скорочують гнучкість програм.

У доповіді наводиться детальний аналіз методів обфускації, їх недоліки та переваги, а також презентується програмна реалізація системи утруднення декомпіляції коду програмного забезпечення.

Список літератури

1. Касперски К. Обфускация и ее преодоление. [Електронний ресурс] – www.dellstroy.homeip.net.
2. Методы обфускации. Журнал "Хакер", Спец. вып. № 66, 2009. – С. 8-13.
3. Смірнов О.А. Основи захисту інформації: Навч. пос.. / А.О. Смірнов, Л.Г. Віхрова, С.І. Осадчий, В.Ю. Ковтун, Є.В. Мелешко. – Кіровоград: РВЛ КНТУ, 2011. – 322 с.

Огляд технології розробки програмного забезпечення Rational Unified Process (RUP)

**Н.В. Підгорна, студент,
О.К. Коноплицька, асистент**

Кіровоградський національний технічний університет

На сьогоднішній день у світі існує величезна кількість процесів для створення ПЗ. Тим не менш, саме технології, що розглядають повний життєвий цикл проекту розробки ПЗ, що поєднують в собі науковий підхід, серйозну базу досліджень і мають історію реального використання та адаптації, відносно небагато. З методологій і технологій, які отримали певне визнання на даний момент, можна назвати наступні: Datarun, CMM, Microsoft Solution Framework (MSF), Oracle Method, Rational Unified Process (RUP), SADT (IDEF_x).

Особливе місце в цьому списку займає технологія Rational Unified Process (RUP), яка претендує на роль світового корпоративного стандарту. RUP представляє собою, програмний продукт, розроблений компанією Rational Software, яка зараз знаходиться в складі IBM.

Основними принципами RUP технології є:

- 1) Ітераційний та інкрементний (нарощуваний) підхід до створення ПЗ.
- 2) Планування і управління проектом на основі функціональних вимог до системи - варіантів використання.
- 3) Побудова системи на базі архітектури ПЗ.

Перший принцип є визначальним. Відповідно до нього розробка системи виконується у вигляді декількох короткострокових міні-проектів фіксованої тривалості (від 2 до 6 тижнів), званих ітераціями. Кожна ітерація включає свої власні етапи аналізу вимог, проектування, реалізації, тестування, інтеграції і завершується створенням працюючої системи.

Ітераційний цикл ґрунтується на постійному розширенні та доповненні системи в процесі декількох ітерацій з періодичним зворотнім зв'язком і адаптацією модулів, які додаються до існуючого ядра системи. Система постійно розростається крок за кроком, тому такий підхід називають ітераційним та інкрементним.

Згідно RUP, розробка ПЗ розбивається на окремі цикли, в кожному з яких створюється нове покоління продукту. Кожен цикл, в свою чергу, розбивається на чотири послідовні стадії:

- початкова стадія (inception);
- стадія розробки (elaboration);
- стадія конструювання (construction);
- стадія введення в дію (transition).

Кожна стадія завершується в чітко визначеній контрольній точці (milestone). В цей момент часу повинні досягатися важливі результати і прийматися критично важливі рішення про подальшу розробку.

Початкова стадія може приймати безліч різних форм. Для великих проектів початкова стадія може вилитися у всебічне вивчення всіх можливостей реалізації проекту, яке може тривати місяці. Під час початкової стадії виробляється бізнес-план проекту - визначається, скільки приблизно він буде коштувати і який дохід принесе. Визначаються також межі проекту, і виконується деякий початковий аналіз для оцінки розмірів проекту.

Результатами початкової стадії є:

- загальний опис системи: основні вимоги до проекту, його характеристики та обмеження;
- початкова модель варіантів використання (ступінь готовності - 10-20%);
- початковий проектний глосарій (словник термінів);
- початковий бізнес-план;
- план проекту, що відображає стадії та ітерації;
- один або кілька прототипів.

На стадії розробки виявляються більш детальні вимоги до системи, виконується високорівневий аналіз предметної області та проектування для побудови базової архітектури системи, створюється план конструювання та усуваються найбільш ризиковані елементи проекту.

Результатами стадії розробки є:

- модель варіантів використання (завершена принаймні на 80%), яка визначає функціональні вимоги до системи;
- перелік додаткових вимог, включаючи вимоги не функціонального характеру і вимоги, не пов'язані з конкретними варіантами використання;
- опис базової архітектури майбутньої системи;
- працюючий прототип;
- уточнений бізнес-план;
- план розробки всього проекту, що відображає ітерації та критерії оцінки для кожної ітерації.

Найважливішим результатом стадії розробки є опис базової архітектури майбутньої системи. Ця архітектура включає:

- модель предметної області, яка відображає розуміння бізнесу і служить відправним пунктом для формування основних класів предметної області;
- технологічну платформу, яка визначає основні елементи технології реалізації системи та їх взаємодію.

Ця архітектура є основою всієї подальшої розробки, вона служить свого роду проектом для наступних стадій. Надалі неминучі незначні зміни в деталях архітектури, проте, серйозні зміни мало ймовірні.

Стадія розробки займає близько п'ятої частини загальної тривалості проекту. Основними ознаками завершення стадії розробки є дві події:

- розробники в змозі оцінити з достатньо високою точністю, скільки часу буде потрібно на реалізацію кожного варіанту використання;

- ідентифіковані всі найбільш серйозні ризики і відомо, як впоратися з ними.

Сутність планування полягає у визначенні послідовності ітерацій конструювання і варіантів використання, реалізованих на кожній ітерації. Ітерації на стадії конструювання є одночасно інкрементними і повторюваними:

- ітерації є інкрементними відповідно до тієї функцією, яку вони виконують. Кожна ітерація додає чергові конструкції до варіантів використання, які реалізовані під час попередніх ітерацій;

- ітерації є повторюваними по відношенню до коду, що розробляється. На кожній ітерації деяка частина існуючого коду переписується з метою зробити його більш гнучким.

Результатом стадії конструювання є продукт, готовий до передачі кінцевим користувачам. Як мінімум, він містить наступне:

- ПО, інтегроване на необхідних платформах;
- інструкція користувача;
- опис поточної реалізації.

Призначенням стадії введення в дію є передача готового продукту в розпорядження користувачів. Дана стадія включає:

- бета-тестування, що дозволяє переконатися, що нова система відповідає очікуванням користувачів;

- паралельне функціонування з існуючою (legacy) системою, яка підлягає поступовій заміні;

- конвертація баз даних;
- оптимізацію продуктивності;
- навчання користувачів і фахівців служби супроводу.

Статичний аспект RUP представлений чотирма основними елементами: ролі, види діяльності, робочі продукти, дисципліни.

Поняття "роль" (role) визначає поведінку і відповідальність особистості або групи особистостей, що складають проектну команду. Одна особистість може грати в проекті багато різних ролей.

Під видом діяльності конкретного виконавця розуміється одиниця виконуваної ним роботи. Вид діяльності (activity) відповідає поняттю технологічної операції. Він має чітко визначену мету, зазвичай виражається в термінах отримання або модифікації деяких робочих продуктів (artifacts), таких, як модель, елемент моделі, документ, вихідний код або план. Кожен вид діяльності пов'язаний з конкретною роллю. Тривалість виду діяльності становить від декількох годин до декількох днів, він зазвичай виконується одним виконавцем. Будь-який вид діяльності повинен бути елементом процесу планування. Прикладами видів діяльності можуть бути планування ітерації, визначення варіантів використання і діючих осіб, виконання тесту на продуктивність. Кожен вид діяльності супроводжується набором настанов (guidelines), що представляють собою методики виконання технологічних операцій.

Дисципліна (discipline) відповідає поняттю технологічного процесу і являє собою послідовність дій, що приводить до отримання значимого результату.

В рамках RUP визначено шість основних дисциплін:

- бізнес-моделювання;
- вимоги;
- аналіз і дизайн;
- реалізація;

- тестування;
- постачання (Deployment);
- і три допоміжних:
- управління конфігурацією і змінами;
- управління проектом;
- створення інфраструктури.

На мою думку, RUP - це найкращий старт, який може зробити організація на шляху до успішного процесно-орієнтованого ведення проектів з розробки ПЗ.

Список літератури

1. А.М. Вендров Современные технологии создания программного обеспечения - Jet Info Online, #4/2004
2. Ф. Крачтен Введение в Rational Unified Process.: Пер. с англ. - М.: Вильямс, 2002

Програмне забезпечення системи автентифікації користувача ЕОМ

М.С. Козіров, студент,
С.В. Михайлов, асистент

Кіровоградський національний технічний університет

Стрімкий розвиток засобів захисту інформації й методів їхнього аналізу в ряді випадків приводить до зміни погляду на безпеку існуючої апаратури захисту інформації, перегляду моделі порушника й підвищення нормативних вимог. Найбільшою мірою зазначені зміни характерні для підсистем автентифікації, у яких виникає необхідність забезпечення інформаційної безпеки в умовах взаємної недовіри або змови учасників протоколу, що дозволяє говорити про критичні підсистеми автентифікації. Найважливішим етапом створення таких підсистем є проектування алгоритмів автентифікації.

На сучасному рівні розвитку обчислювальної техніки кожний користувач інформаційних систем зіштовхується із процедурами: "ідентифікація" і "автентифікація", неодноразово протягом робочого дня. Ці процедури виконуються щораз, коли користувач уводить пароль для доступу до комп'ютера, у мережу, до бази даних або при запуску прикладної програми. У результаті їхнього виконання він одержує або доступ до ресурсу, або відмову в доступі. Строго кажучи, цей процес складається із двох частин – ідентифікації й автентифікації. Ідентифікація – це пред'явлення користувачем якогось унікального, властивого тільки йому ознаки-ідентифікатора. Це може бути пароль, якась біометрична інформація, наприклад відбиток пальця, персональний електронний ключ або смарт-карта й т.і. Автентифікація – це процедура, що перевіряє, чи має користувач із пред'явленим ідентифікатором право на доступ до ресурсу. Ці процедури нерозривно зв'язані між собою, оскільки спосіб перевірки визначає, яким образом і що користувач повинен пред'явити системі, щоб одержати доступ.